A Spectrum of IV&V Modeling Techniques

Definition of Error-Seeding Methods for Modeling Languages[1]

Only final corrected versions of models are available, but we want to compare test engines' ability to detect errors, so we need some way of "seeding" correct models with errors. For a given system, we have 1) a model of the implementation, and 2) a set of properties to be verified. So two error-seeding approaches are possible: we can alter the model of the implementation, or we can alter the property specification. Since the models we deal with are produced automatically by translation tools ($RSML^{-e}$ is automatically translated to PVS, SMV or finite-state machines) mutated models can be produced by adding randomized error-injection into the translation tools. Property specifications are also translated automatically from temporal logic queries using automated tools, and so a similar randomized error-injection approach can also be used to produce faulty property specifications. In addition, we can simply test for properties we know or suppose not to be true.

1. University of Minnesota (UMN) Modeling Languages and Test Engines

Detailed information concerning UMN's $RSML^{-e}$ language, related tools, and past case studies can be found in the "Case Study" report associated with this project, or the following documents: "Collect Models from UMN Clients" (umn_models.pdf), "Definition of UMN Languages: $RSML^{-e}$" (umn_languages.pdf), and "Definition of UMN Test Engines" (umn_test_engines.pdf).

2. West Virginia University (WVU), NASA IV&V Modeling Languages and Test Engines

Detailed descriptions of WVU / NASA languages and models can also be found in the "Case Study" report, or: "Definition of WVU Languages" (wvu_languages.pdf) and "A Model-Base Approach to Reactive, Self-Configuring Systems" (livingstone.pdf). WVU / NASA's partial random search test engine, to be evaluated in this project, is described in "Definition of WVU Test Engines" (wvu_test_engines.pdf). For more information see

---

[1] Because this document was delivered behind schedule, it contains information up-to-date 10/19/02, there is significant overlap with the documents "Definition of Validation Methods for Modeling Languages" and "Definition of Metrics Collection Methods" delivered at the same time.

also "An Alternative to Model Checking: Verification by Random Search of AND-OR Graphs Representing Finite-State Models" (alternative.pdf).